(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl. 6
G06F 15/16

(11) 공개번호 특1998-070410
(43) 공개일자 1998년10월26일

(21) 출원번호 특1998-000332
(22) 출원일자 1998년01월09일

(30) 우선권주장 8/780,8231997년01월09일미국(US)
(71) 출원인 선마이크로시스템스,인코퍼레이티드 올센 케니쓰
미국, 캘리포니아 94303, 팔로 알토, 산 안토니오 로드 901
(72) 발명자 나가래트남나타라
미국, 뉴욕 13210, 시라쿠스, 아파트.1, 620 유클리드 애비뉴
바이른스티븐비.
미국, 캘리포니아 95118, 산 조세, 5269 카터 애비뉴
(74) 대리인 강명구

심사청구 : 없음

(54) 시스템 자원으로의 소프트웨어 접근을 제어하기 위한방법 및 그 장치

요약

컴퓨터 네트워크를 통해 서버 컴퓨터와 통신하는 클라이언트 컴퓨터에서 선택된 소프트웨어를 설치하고 동작시키기 위한 방법, 시스템 및 소프트웨어가 설명된다. 본 발명의 한 특징에서 운영체제(동작 시스템)를 실행시키는 컴퓨터에서 사용되는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근정도를 제어하기 위한 방법이 제공된다. 운영체제 자원으로의 접근정도가 소프트웨어 프로그램을 위해 규정되며, 소프트웨어 프로그램을 실행하기 위한 지시들을 포함하는 적어도 하나의 파일이 서버 컴퓨터로부터 상기 컴퓨터로 적재된다. 소프트웨어 프로그램이 상기 컴퓨터에 의해 실행되는때 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도를 결정하기 위해 파일이 시험된다. 소프트웨어 프로그램이 실행되고, 상기 소프트웨어 프로그램과 관련된 프로그램 지시가 소프트웨어가 컴퓨터에서 실행되고 있는때 인터럽트된다. 상기 프로그램 지시가 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도 바깥에 있는 동작을 포함하는지에 대해 결정이 있게 되며, 상기 소프트웨어 프로그램이 상기 소프트웨어 프로그램이 상기 소프트웨어에 이용될 수 있는 시스템-수준 접근 정도내에 있는 상기 컴퓨터와 관련된 시스템-수준 자원에 접근하는 것이 허용되는때 상기 프로그램 지시가 실행된다.

대표도

도1

명세서

도면의 간단한 설명

도 1a 는 인터넷을 통해 컴퓨터 네트워크에 의해 사용자와 인트러넷 모두가 결합되는 광지역 컴퓨터 네트워크를 나타낸 도면.

도 1b 는 본 발명의 실시예에 따라 클래스 파일의 수집을 도식적으로 도시한 도면.

도 2b 는 본 발명의 실시예에 따라 보관 파일 자료 포맷의 도식적 도면을 도시한 도면.

도 3a 는 본 발명의 실시예에 따라 고객측 디렉토리 구조를 개략적으로 도시한 도면.

도 3b 는 본 발명의, 실시예에 따라 고객측 구성 파일 구조를 개략적으로 도시한 도면.

도 3c 는 본 발명의 실시예에 따라 고객측 접근 파일 구조를 개략적으로 도시한 도면.

도 3d 는 본 발명의 실시예에 따라 고객측 그룹 스펙 파일의 구조를 개략적으로 도시한 도면.

도 4 는 본 발명의 실시예에 따라 자원에 접근하기 위한 요구를 실행하는 방법을 설명하는 처리 흐름도.

도 5 는 본 발명의 실시예에 따라 클래스 파일을 유효하게 함과 관련된 단계들을 설명하는 처리 흐름도.

도 6 은 본 발명의 실시예에 따라 애플릿을 실행함과 관련된 단계들을 설명하는 처리 흐름도.

도 7 은 본 발명의 실시예에 따라 보안 관리자를 호출함과 관련된 단계를 설명하는 처리 흐름도.

도 8 은 본 발명에 따라 컴퓨터 시스템을 개략적으로 도시한 도면.

* 부호설명

50 ... 인트러넷 60 ... 서버

62, 64 ... 클라이언트(client) 302 ... 사용자 디렉토리

102, 104 ... 인트러넷 106 ... 사용자

108 ... 인터넷 110, 114 ... 게이트웨이

118 ... 고용량 자료라인 120, 124 ... 게이트웨이

302 ... 사용자 디렉토리 304 ... 브라우저 디렉토리

306 ... 특성 파일 308 ... 사용자 선호 아이템

310 ... 영상 자료 블록 312 ... 구성 파일 이름

314 ... 그룹 명세서 파일 이름 356 ... 접근 파일 이름

360 ... 접근 파일 362 ... 프린시플(principal)

812 ... 네트워크 연결장치 830 ... 컴퓨터 시스템

832 ... 프로세서 834 ... 저장장치

838 ... 대량 기억장치 840 ... 입력/출력장치

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야 종래기술

본 발명은 컴퓨터에서 실행되는 소프트웨어에 의해 컴퓨터 자원으로의 접근을 제어하기 위한 방법 및 장치에 대한 것이다.

특히 본 발명은 서버 컴퓨터로부터 클라이언트 컴퓨터로 다운로드되는 소프트웨어에 의해 클라이언트 컴퓨터에서의 시스템 자원으로의 접근을 제어하기 위한 방법 및 장치에 관한 것이다.

개인용 컴퓨터(PC)가 발생하기 전에는, 컴퓨터 사용자는 명령을 입력시키기 위한 키이보드와 출력을 관찰하기 위한 비디오 표시장치(또는 프린터)를 포함하는 단말기를 사용하는 큰 메인프레임 컴퓨터에서 실행되는 소프트웨어를 동작시키도록 제한되었다. 비록 메인 프레임이 매우 위력있는 계산 플랫폼을 제공하였으나, 이들에게는 심각한 단점이 있었다. 특히, 메인 프레임은 설치하고 동작시키기가 비싸며, 이들은 모든 사용자가 한 단말기를 통해 메인프레임에 직접 연결되도록 요구하였으며, 이는 많은 사람이 메인 프레임으로 접근하는 것을 제한하였다. 또한 사용자는 이들의 계산 환경에 매우 제한된 제어만이 가능했으며, 이들의 작업 스타일과 문제를 메인프레임 컴퓨터 소프트웨어와 관리에 적합하도록 적용해야만 했다.

1970년대 말에 시작한 개인용 컴퓨터는 개인용, 사업용 및 과학용 모두로서 지배적인 계산 플랫폼으로서 메인프레임을 대체하기 시작했다. 싱글 사용자를 위해서도, 개인용 컴퓨터는 많은 처리 일들을 동시에 수용해야하는 오래된 메인 프레임과 같은 계산 속도를 제공할 수 있었다.

또한, 개인용 컴퓨터에서 실행되었던 소프트웨어는 더욱더 사용자와 친숙해지게 되었으며, 이에 의해서 컴퓨터 사용자로 하여금 컴퓨터와 소프트웨어 모두가 이들의 특정 계산 요구에 적합하게 적용시킬 수 있도록 허용한다. 단말기로부터 메인프레임으로의 연결을 필요로했던 점이 해제됨으로써 개인용 컴퓨터는 조직 또는 가정내 어느 곳에서도 개인용 컴퓨터가 위치할 수 있도록 하였다. 이같은 능력은 계산 능력이 필요한 장소에 위치할 수 있기 때문에 개인용 컴퓨터의 메인 프레임에 대한 우세한 지배를 보장하였다. 사용자는 더 이상 덩치가 크고 값비싼 메인프레임 계산 센터들에서 이들의 동작을 조작해야할 필요가 없게되었다.

개인용 컴퓨터의 계산력과 자료저장 능력은 1980년대에 들어 크게 개선되었기 때문에 개인용 컴퓨터의 지배가 확보된 것으로 파악된다. 그러나 1980년대가 마감된 때와 새로운 현상이 지난 20년의 개인용 컴퓨터 혁신을 이어받아 나타나는 것으로 파악되기 시작했다. 오늘날 전례없던 많은 수의 개인용 컴퓨터는 고속자료 네트워크를 통해 서로 연결되어 있다. 현재 가장 널리 알려진 네트워크는 인터넷이다. 이 인터넷은 전세계적으로 다양한 사업, 학원 그리고 개인용 컴퓨터로 구성된 네트워크이다. 인터넷의 인기, 특히 World wide Web이라 하는 인터넷의 특징은 많은 조직으로 하여금 인트러네트라 불려지는 내부 컴퓨터 네트워크를 형성하도록 촉진해왔다. 네트워크 컴퓨터에서의 이같은 관심은 고속 자료 네트워크와 점점 복잡해져가는 네트워크 서버, 라우터(router), 그리고 많은 독립된 개인용 컴퓨터가 효율적으로 통신하도록 하는 다른 장치의 조합에 의해 고조되었다.

World wide web 의 매력은 높은 비쥬얼 특징에 있으며, 이는 개인용 컴퓨터의 메인 프레임에 대한 우세에도 큰 역할을 하였다. 대개, World wide web 는 브라우저(browser)를 실행시키는 클라이언트 컴퓨터로 자료를 전송시키는 서버를 포함하는 다양한 웹브 사이트로 조직된다. 상기 브라우저는 사용자에게 윈도우와 다양한 제어를 제공하는 소프트웨어이며, 상기 제어를 통해 서버로 부터의 자료가 접근될 수 있다. World wide web 자료의 특히 유용한 특징은 사용자가 마우쓰 버튼의 사용과 같은 매우 간단한 명령을 통해 한 서류에서 다른 한 서류로 그리고 심지어는 한 웹브 장소로 부터 다른 한 웹브장소로 신속하게 이동할 수 있도록 하이퍼 텍스트 명령을 통해 연결되도록 하는 능력이다. World wide web 를 사용하여, 사용자는 텍스트, 그래픽에 접근하거나 다운로드시킬 수 있고, 전세계적으로 퍼져있는 장소로부터 사운드를 들을 수 있다. 또한, 사용자는 클라이언트 컴퓨터에 이미 설치된 프로그램을 수정할 수 있는 새로운 소프트웨어를 다운로드 시킬 수 있다.

인터넷을 통한 워드 와이드 웹브의 사용자가 이용할 수 있는 상기의 특징들은 인터넷에 유사하게 배치된 클라이언트와 서브를 포함하는 인트러넷, 비-공공 컴퓨터 네트워크를 통해 로컬 네트워크 사용자에게 제공될 수 있다. 이같은 능력은 이들의 역할을 수행하는 고용인에게 유용한 정보가 네트워크를 통해서 조직내 개인용 컴퓨터로 신속하게 분산될 수 있기 때문에 많은 조직으로부터 더욱더 많은 주목을 받고 있다. 특히, 많은 조직들은 데이터 베이스로의 접근을 제공하기 위해 인트러넷을 사용하고 있으며, 그와 같은 인트러넷을 사용하는 조직내 개인을 위해 컴퓨터 소프트웨어 프로그램을 사용하고 있다. 일례로서, 자바

TM 프로그래밍 언어(캘리포니아, 팔로 알토의 선 마이크로시스템스로 부터 구입가능)를 사용하여 발생된 고객 소프트웨어 애플릿(applets)은 원격 컴퓨터상에 앞서 설치된 소프트웨어 및 자료와 협력하여 동작될 수 있으며, 이때 상기 컴퓨터는 인트러넷 외부 또는 내부에 위치하여 종래에서와 같이 특수목적 소프트웨어를 다수 복사본을 퍼뜨리고 유지시키는 번거로움 없이 이들의 작업에 특정된 자료와 소프트웨어에 사용자가 접근할 수 있도록 한다.

안전한 인트러넷을 통해 분산된 소프트웨어가 고객 컴퓨터의 시스템 자원으로 완전히 접근할 수 있도록 하는 것이 바람직하

며, 인트러넷 시스템 외부의 덜 안전한 네트워크를 통해 분산된 소프트웨어는 파일 이동 능력과 같은 시스템 리소스(자원)으로의 접근이 허용되지 않는때, 이는 그와 같은 소프트웨어가 항상 신뢰될 수 있는 것은 아니므로 그렇다. 가령, 몇가지 소프트웨어 응용은 호스트 컴퓨터를 통해 컴퓨터 바이러스를 설치하는 기능을 포함한다. 다른 소프트웨어가 일정 자원에 접근하는 것은 제한하면서 신뢰된 소프트웨어가 같은 자원에 접근하도록 하는 방법 또는 장치가 아직까지는 존재하지 않는다. 따라서 사용자는 모든 소프트웨어(신뢰되든 그렇지 않든)가 모든 시스템 자원에 접근하게 하는 것과 모든 소프트웨어의 접근을 제한하는 것 사이에서 번갈아 사용하도록 하여 고객 시스템의 보안을 보전하여야 한다.

따라서, 이는 다수의 컴퓨터 사용자가 컴퓨터 네트워크를 통해 연결되는 조직내에서 특히 유익하며, 매우 분산되어 있는 컴퓨터 네트워크와 관련되어 있는 상기 설명된 문제가 완화될 수 있도록 네트워크를 통한 정보와 소프트웨어 모두를 위한 자원 접근을 제어하기 위한 방법과 시스템을 제공하도록 한다. 하기에서 설명되는 바와 같이 본 발명을 상기의 요구사항을 해결하는 것이다.

발명이 이루고자하는 기술적 과제

본 발명은 컴퓨터 네트워크에서 서버 컴퓨터로 작용하는 두 번째 컴퓨터와 통신하는 고객 컴퓨터로서 작용하는 첫 번째 컴퓨터에서의 선택된 소프트웨어 응용에 의해 서버 자원으로의 접근을 제어하기 위한 방법, 시스템 및 소프트웨어를 제공하므로써 네트워크된 컴퓨터에서 분산을 관리하는데 상기 설명된 어려움을 해결한다.

본 발명의 한 특징에서, 컴퓨터에서 실행되는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하기 위한 방법이 제공된다.

운영체제 자원으로의 접근정도는 소프트웨어 프로그램을 위해 정의되며, 소프트웨어 프로그램을 실행하기 위한 지시를 포함하는 적어도 하나의 파일이 컴퓨터상에 적재된다. 상기의 파일은 소프트웨어 프로그램이 컴퓨터에 의해 실행되는때 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근의 정도를 결정하기 위해 시험된다. 소프트웨어 프로그램이 실행되며, 소프트웨어 프로그램과 관련된 자원을 확보하기 위해 접근을 요구하는 프로그램 지시는 소프트웨어가 컴퓨터에서 실행되고 있는때 인터셉트된다. 만약 프로그램 지시가 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도 바깥인 동작을 포함하고, 소프트웨어 프로그램이 상기 소프트웨어에 이용될 수 있는 시스템-수준 접근 정도 이내에 있는 컴퓨터와 관련된 시스템-수준 자원에 접근할 수 있다고 결정되며, 상기 프로그램 지시가 실행된다.

본 발명의 또다른 실시예에서, 운영체제를 실행시키는 고객 컴퓨터에서 실행되는 소프트웨어 프로그램을 위한 시스템 자원의 접근 정도를 제어하는 방법이 제공되며, 이때 적어도 몇 개의 운영체제 자원이 고객 컴퓨터에 연결된 서버 컴퓨터 상에 있게 된다. 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도가 규정되며, 고객 컴퓨터에서의 소프트웨어 프로그램을 실행하기 위한 운영체제 자원으로의 접근 정도가 규정되며, 고객 컴퓨터에서의 소프트웨어 프로그램을 실행하기 위한 지시를 포함하는 적어도 하나의 파일이 적재된다. 소프트웨어 프로그램이 고객 컴퓨터에 의해 실행되는때 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근의 정도를 결정하기 위해 파일이 시험된다. 소프트웨어 프로그램을 고객 컴퓨터상에서 실행되며, 소프트웨어 프로그램이 고객 컴퓨터에서 실행되는때 소프트웨어 프로그램과 관련된 프로그램 지시가 해석된다. 프로그램 지시가 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도외에 있는 동작을 포함하는 가에 대한 결정이 있게되며, 상기 시스템 프로그램이 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준접근 정도내에 있는 시스템-수준 자원에 접근할 수 있음이 결정되는때 상기 프로그램 지시가 실행된다.

발명의 구성 및 작용

애플릿(applets)에 의해 시스템 자원으로의 접근을 제어하기 위한 방법 및 장치의 일정 실시예가 첨부도면을 참조하여 하기에서 설명될 것이다.

본 발명에 따른 한 네트워크의 설명이 도 1a 에서 제공된다. 도 1a 에서 설명된 네트워크내에는 인트러넷(102)(104)과 (106)으로 도시된 개인용 컴퓨터가 포함된다. 인트러넷(102)(104)의 구조는 도 1b 와 관련하여 하기에서 더욱더 상세히 설명된다. 인트러넷과 사용자 모드는 다양한 컴퓨터 게이트웨이(G/W)를 통해 컴퓨터 네트워크에 연결된다. 어떤 실시예에서는 컴퓨터 네트워크가 인터넷을 포함한다. 특히 도 1a 와 관련하여서는 인트러넷(102)이 (108)로 도시된 인터넷을 통해서 인트러넷(104)과 사용자(106)에 결합된다. 인트러넷(102)과 인터넷(108)사이의 연결은 인트러넷(102)과 결합된 게이트웨이(110)와 고용량 자료라인(112)을 통해서 먼저 제공된다. 고용량 라인(112)으로 부터의 자료는 게이트웨이(114)를 통해 인터넷(108)으로 보내지며, 이 자료는 두 번째 게이트웨이(116)를 통해 (118)로 도시된 고용량 자료라인으로 보내진다. 컴퓨터 네트워크 기술분야에서 숙련된 자에 의해 알 수 있는 바와 같이, 자료라인(118)은 자료라인(112)과 같을 수 있으며, 혹은 여러 다른 사용자 및 네트워크가 결합되는 분리된 한 고용량 자료라인을 나타내기도 할 수 있다.

인터넷(108)을 통해 고속 자료라인(118)을 통과한 인트러넷(102)으로 부터 이동하는 자료는 게이트웨이(120)를 통해 인트러넷(104)으로 보내지거나 게이트웨이(124)를 통해 사용자(106)에게도 보내진다. 따라서, 설명된 실시예에 따라, 자료가 사용자(106), 인트러넷(104), 그리고 인트러넷(102) 가운데에서 통과될 수 있다. 특히, 자료는 상기에서 설명된 바와 같이 인터넷(108)을 통해 이동할 수 있거나, 또는 사용자(106)와 인트러넷(104)사이의 자료라인(118)을 통과할 수 있다. 어떤 실시예에서는, 인트러넷(104) 및 인트러넷(102)이 익스트러넷과 같은 종래 기술분야에서 알려진 네트워크 구조를 통해 직접 연결될 수 있기도 하다. 익스트러넷은 네트워크 배열로서, 정해진 네트워크 또는 개인이 지정된 자료 전송을 통해 원격 네트워크에 결합된다. 이같은 연결은 도 1a 에서 도시된 바와 같이 인터넷을 통해 경로가 정해진 자료를 포함할 수 있으며, 혹은 ISDN 또는 T-1 자료라인을 통하는 것과 같은 직접적인 자료공급이 될 수 있다. 상기 구성을 만들기 위한 방법 및 재료외에도 다양한 구성이 컴퓨터 네트워크 및 통신 기술인 당해분야에 숙련된 자에 의해 명백해질 수 있다.

도 1a 에서(102) 또는 (104)로 설명된 바와 같은 인트러넷의 한 실시예가 도 1b 에서 (50)으로 제공된다. 한 전형적인 인트러넷(50)은 클라이언트(고객)(62)(64)에 결합된 서버(60)를 포함한다. 또한, 서버(60)는 라우터, 허브 또는 노드(68)에서 도시된 바와 같은 유사한 자료전달장치를 통해 (70)(72) 및 (74)에서 도시된 바와 같은 다른 컴퓨터에 결합될 수 있다. 또한, 원격한 클라이언트(도시되지 않음)는 직접적인 라인 또는 모뎀 또는 유사한 장치를 사용하는 전화선을 통해서 서버(60)에 연결될 수 있다. 어떤 경우에서는, 인트러네트(50)로의 접근은 박스(75)에 의해 설명되는 방화벽 구성에 의해 고도로 제어될 것이다. 원격 클라이언트(78)와 같은 방화벽 외부에 있는 사용자로부터의 통신 설정은 보호된 서버로의 접근을 허용하는 게이트웨이를 횡단하므로써 달성될 수 있다. 이와 같은 게이트웨이가 (76)으로 설명된다.

대개, 하나의 서버는 직접 또는 라우터와 같은 장치를 통해서 서버와 통신하게 되는 다양한 클라이언트에게 접근될 수 있는 자료 및 소프트웨어를 제공한다. 서버, 라우터 및 다양한 클라이언트 머신의 구조, 유지 및 동작은 당해 기술분야에 잘 알려져 있다. 어떤 특정 실시예에서는 서버(60)가 워드 와이드 웹브에서 자료를 관찰하도록 사용되는 것과 같은 브라우저 소프트웨어와 호환성이 있는 자료를 제공하도록 구성될 것이다. 특히, 서버(60)에 의해 제공되는 자료는 전형적인 브라우저 소프트웨어를 사용하여 시험될 수 있는 자료의 페이지 형태가 될 것이다. 한 실시예에서, 서버와 클라이언트는 캘리포니아, 팔로 알토에 소재하는 선 마이크로시스템스로부터 구입될 수 있는 자바

TM 프로그래밍 언어로 기록된 것과 같은 애플릿의 형태로 자료뿐 아니라 컴퓨터 소프트웨어를 상호교환하도록 구성된다. 본 원 명세서에서 사용되는바 애플릿은 대개 서버와 같은 소스 컴퓨터로부터 클라이언트 머신으로 보내지도록 된 소프트웨어 프로그램이며 클라이언트 컴퓨터에 이미 설치된 소프트웨어와 협력하여 실행된다. 한 실시예에서, 상기 소프트웨어는 이와 함께 상기 애플릿이 실행되는 것으로서 상기 설명된 브라우저 소프트웨어일 수 있다. 대개, 애플릿은 브라우저 소프트웨어 자체가 수행할 수 있도록 구성되지 않은 다양한 계산 작업을 수행함으로써 브라우저 소프트웨어로 추가의 기능을 제공한다. 따라서, 애플릿들을 다운로드시킨 사용자는 브라우저 소프트웨어에 추가의 기능을 제공하는데 이는 그렇지 않았더라면 브라우저 소프트웨어로 제공될 수 없었던 기능이다. 이같은 추가의 능력으로는 데이타베이스로의 고객 인터베이스를 더욱더 포함한다.

일반적으로, 클라이언트(62)는 사용자 에이전트 또는 브라우저를 사용하여 서버(60)로 연결된다. 사용자 에이전트로는 선 마이크로시스템 사로부터 구입되는 HotJava

TM, 네트스케이프 커뮤니케이션스 코포레이션 사로부터 구입되는 Netscape등이 있다. 한 실시예에서, 사용자 에이전트는 일반적으로 애플릿 코드를 실행하는 처리 엔진, 애플릿이 일정 시스템 자원으로의 접근을 갖는지에 대한 결정에서 사용된 보안 관리자를 포함한다. 이같은 보안 관리자의 예가 하기에서 제공된다.

한 실시예에 따라, 인터넷상에 또는 인트러넷내에 위치하는 서버는 애플릿을 규정하는 클래스 파일을 포함하는 클래스 라이브러리를 제공한다. 이같은 클래스 라이브러리의 한 예가 자바

TM 클래스 라이브러리이다. 특히, 서버는 애플릿을 구성시키는 클래스 파일과, 애플릿을 나타내는 HTML 코드 포함하는 특정 Web(웨브) 페이지들을 담고 있을 수 있다.

본 발명의 한 실시예에 따라, 애플릿은 소스 또는 서버 컴퓨터로부터 클라이언트 머신으로 다운로드된 클래스 파일로부터 만들어진다. 클래스 파일들은 한 보관 파일내로 그룹으로 모아진다. 또한, 보관 파일은 디지탈식으로 표시되어 보관 파일들로부터 발생된 애플릿의 근원이 신뢰할 수 있도록 결정될 수 있다. 다음에 보관 파일의 표시가 수정되어 어느 시스템 자원이 애플릿이 실행되고 있는 자원으로 접근될 수 있는가를 결정하도록 한다. 표시를 사용하므로써 애플릿이 발생되는 서버의 보안 상태를 참조하므로써 제어되어질 애플릿에 대한 클라이언트 머신의 시스템 자원으로의 접근을 가능하게 한다. 일례로서 한 클라이언트에서 실행되는 애플릿은 각 클라이언트에서의 애플릿과 관련된 허용이 다르기 때문에 두 번째 클라이언트에서 실행되는 같은 애플릿과는 다른 접근 권한을 갖는다. 따라서, 이같은 자원접근 제어는 상기 자원을 포함하는 머신과 같은 인트러넷내 보안 머신과 관련된 애플릿을 가능하게 하여 인터넷에서의 머신과 같은 보안이 되지 않는 머신과 관련된 애플릿보다는 자원으로의 더욱더 많은 접근을 갖도록 한다.

도 2a 는 본 발명의 실시예에 따라 클래스 파일들의 수집을 나타내는 도식적 도면이다. 서버에서 대개 사용되는 클래스 자료

파일의 수집 포맷은 서명을 받아들이도록 되지 않는다. 즉, 각 클래스 파일은 서버에 있는 한 클래스를 규정한다. 상기 포맷은 상기의 수집이 가령 클래스 1(202), 클래스 2(204), 그리고 클래스 N(206)과 같은 어떠한 수의 클래스도 포함하도록 한다. 한 클래스는 자료와 방법을 규정하는 소프트웨어 구성으로서 구성될 수 있으며, 혹은 뒤에 가서 그같은 클래스로부터 구성되는 어떠한 애플릿에 특정된 자료로 등작되는 명령문의 순서일 수 있다. 다시 말해서, 앞서 언급한 바와 같이, 앞서 규정된 클래스를 실증하므로써 구성될 수 있다.

한 애플릿의 실행은 시스템 자원으로 접근하기 위한 요구 또는 명령을 수반한다.

애플릿이 많은 각기 다른 시스템 자원에 접근하기 위한 지시를 포함하고 있지만, 보안 문제 때문에, 한 애플릿은 명시된 시스템 자원 모두로의 접근이 허용되거나 본 발명 디자인의 제한하에서 명시된 시스템 자원 어느 것으로도 접근이 허용되지 않는다. 상기에서 설명된 바와 같이, 시스템 자원 접근으로 전부 또는 전부의 접근은 인트러넷내 시스템에서 실행되는 애플릿이 공지의 근원(origin) 가운데 신뢰되며(trusted), 인트러넷 시스템으로 가는 동등한 애플릿이 보안되지 않은 것으로 간주된다는 점에서 바람직하지 않을 수 있다. 인트러넷 시스템내에서 실행되는 애플릿과 외부로 가는 동등한 애플릿이 대개 시스템 자원으로의 같은 접근권한을 가지기 때문에, 인트러넷 시스템의 보안을 유지시키기 위해 애플릿은 어떠한 접근권한도 받지 않는다.

애플릿이 자원에 접근하는 것을 선택적으로 제어하기 위한 능력은 한 인트러넷 시스템내 사용자가 개별적인 애플릿을 기본으로 하여 자원으로의 접근을 제한하도록 한다. 한 애플릿을 예를 들어 설명하기 위해 사용된 클래스 파일을 갖는 서명(signature) 또는 식별자를 포함하는 것은 인트러넷 조직이 애플릿을 선택적으로 제어할 수 있도록 하는 한가지 방법이다. 발생된 클래스 파일들이 인트러넷 시스템으로 하여금 상기 클래스 파일들로부터 설명된 애플릿과 관련한 적절한 접근권한을 결정할 수 있도록 하는가를 결정할 수 있도록 클래스 파일들을 서명 또는 표시한다. 또한 이같이 클래스 파일들을 서명하는 것은 한 클래스 파일이 간섭을 받았는가에 대한 결정이 될 수 있도록 한다. 한 그룹의 클래스 파일이 디지탈식으로 표시될 수 있도록 하는 한 보관 파일 구조가 도 2b 와 관련하여 하기에서 설명될 것이다.

디지탈식으로 표시될 수 있는 한 보관 파일을 제공하므로써, 보관 파일과 관련된 클래스 파일로부터 구성된 인트러넷 시스템 내부 및 외부의 애플릿이 상기 인트러넷 시스템내 선택된 시스템자원에 접근하도록 하는 것이 가능하게 된다. 보관 파일의 디지탈 서명 또는 표시를 검사하는 것은 주어진 애플릿이 간섭을 받았는지, 그리고 어느 컴퓨터가 상기 애플릿을 표시하였는지를 결정함을 가능하게 한다. 마찬가지로, 접근 권한은 보안의 또는 신뢰된 호스트(컴퓨터)로 부터 혹은 보안되지 않은 호스트로부터 상기 애플릿이 발생되었는가를 바탕으로 하여 접근 권한을 할당받을 수 있다. 또한, 어떤 실시예에서는, 접근 권한의 할당이 사용자가 어느 호스트가 신임을 받는지 그리고 어느 호스트는 신임을 받지 않는지를 결정할 수 있도록 한다.

도 2b 는 본 발명의 실시예에 따른 보관 파일 자료 포맷의 개략적인 도면을 도시한 것이다. 설명된 실시예에서, 보관 포맷은 자바

TM 보관(JAR) 포맷이다. 보관, 또는 보관 파일(210)은 헤더(header) 서명(212)을 포함하는데, 이 서명(표시)은 대개 보관 파일(210)의 유효성을 검증하고 보관 파일(210)에서 이용될 수 있는 접근수준을 결정하기 위해 사용자 에이전트에 의해 사용되는 서명이다. 일반적으로, 헤더 서명(212)은 디지탈 서명으로서, 보관 파일의 크기에 상응하는 정보를 포함하는 다른 정보를 담고있는 범용헤더의 일부분이다. 보관 파일(210)은 예를 들어 클래스 1(202), 클래스 2(204), 그리고 클래스 N(206)와 같은 어떠한 수일 수도 있는 관련 클래스들을 가지며, 이들로부터 애플릿들과 관련된 목적들이 설명된다.

또한, 보관 파일(210)은 자료 블럭(214)과 같은 관련된 자료 블럭을 갖는다. 자료 블럭(214)은 보고나 파일(210)의 일부일 것으로 생각되는 어떠한 임의 자료이거나, 영상 또는 텍스트를 포함할 수 있다. 한 실시예에서, 자료 블럭(214)은 보관 파일(210)과 관련된 클래스(202, 204, 206)들을 설명하는 텍스트들을 포함할 수 있다. 다른 실시예에서는 보관 파일(210)이 자료 블럭을 포함하지 않을 수도 있다.

도 3a 에서, 클라이언트측 디렉토리 구조의 한 실시예가 본 발명에 따라 설명될 것이다. 한 클라이언트를 통해 자원에 접근하고자 하는 요구를 하는 사용자는 사용자 디렉토리(302)와 접속한다. 사용자 디렉토리9302)는 한 브라우저, 또는 사용자 에이전트와 관련된 정보를 포함하는 관련된 브라우저 디렉토리(304)를 갖는다. 상기의 브라우저는 상기에서 언급한 바와 같은 HotJava

TM 브라우저와 같은 어떤 적절한 브라우저일 수 있다. 브라우저 디렉토리(304)는 사용자에 의한 요구에 적절한 한 특성파일(306)을 포함한다. 특성 파일(306)은 대개 사용자에 의해 제공된 브라우저 명세서인 사용자 신호 아이템(308)을 포함한다. 상기 명세서들은 브라우저 설치와 관련된 자료 그리고 상기 브라우저와 관련된 행동 특성을 포함한다.

특성 파일(306)들은 사용자에 의한 특정 요구와 관련된 정보를 더욱더 포함한다. 일례로서, 이같은 정보로는 영상 자료 블럭(310), 구성 파일 이름(312), 그리고 그룹 명세서 파일 이름(314)을 포함한다. 한 실시예에서, 영상 자료 블럭(310)은 상기 요구와 관련된 어떠한 영상도 나타내는 스트링(string)인 자료 파일 이름을 포함한다. 한 구성 파일 이름(312)은 관련된 보안 디스크립터로 요구된 소스의 맵핑을 용이하게 하도록 사용되는 구성파일을 나타내는 스트링이다. 구성 파일의 한가지 예가 도

3b 와 관련하여서 하기에서 설명될 것이다. 그룹 명세서(스펙) 파일 이름(314)은 도 3c 와 관련하여서 하기에서 설명되어질 그룹 명세서 파일을 나타내는 스트링이다.

도 3b 는 본 발명의 한 실시예에 따른 구성파일의 구조를 개략적으로 나타낸 도면이다. 구성 파일(350)은 도 3a 와 관련하여 상기에서 설명된 바와 같은 구성 파일 이름(312)에 의해 표시된 구성 파일의 한 예이다. 상기 구성 파일(350)은 서버, 즉 클라이언트가 접근하고자 하는 서버에서의 자원을 상응하는 접근 파일 이름(356)과 관련시키는 한 테이블(352)을 포함한다. 즉, 테이블(352)은 자원 칼럼(354)내의 엔트리가 접근 파일 이름 칼럼(356)내의 상응하는 엔트리와 관련시킨다. 자원(354)은 가령 파일, 호스트, 그리고 소켓 수와 같은 다양한 시스템은 나타내는 분류기(classifiers)이다. 접근 파일 이름(356)은 해당 접근 파일들을 나타내는데, 이들 파일들은 보안 디스크립터 그리고 접근 파일이 관련되는 시스템 자원으로의 접근 제어와 관련된 다른 정보를 나타낸다.

접근 파일의 구조는 도 3c 와 관련하여 하기에서 더욱 상세히 설명될 것이다. 둘이상의 자원(354)이 같은 보안 디스크립터를 공유한다는 사실 때문에 접근 파일 이름(356) 그리고, 따라서 접근 파일들은 둘이상의 자원(354)과 관련될 수 있음을 이해하여야 한다.

다음에는 도 3c 와 관련하여, 본 발명의 실시예에 따라 접근 파일의 구조가 설명될 것이다. 접근 파일(360)은 프린시플(principals)(362)과 허용(364)을 관련시키는 테이블(361)을 포함한다. 프린시플(362)은 개별적인 호스트 또는 호스트 그룹일 수 있다. 일례로서, 자바.컴은 프린시플(362)인 서버와 같은 개별적인 호스트이다. 선택적으로 자바.컴과 선.컴은 한 그룹인 프린시플(362)을 형성시킨다. 몇가지 실시예에서, 프린시플(362)은 특정 보관 파일의 서명자일 수 있다. 상기 허용(364)들은 보안 디스크립터의 그룹(모음)을 제공한다. 즉, 허용(364)은 허용(364)이 관련되는 프린시플(362)이 접근하게 되는 자원을 지정하는 보안 디스크립터의 그룹(모음)들인 것이다.

도 3d 는 본 발명의 한 실시예에 따른 그룹 명세서(스펙)파일 포맷을 개략적으로 도시한 도면이다. 상기에서 언급한 바와 같이, 도 3a 의 그룹 명세서 파일 이름(314)은 가령 그룹 명세서 파일(370)과 같은 그룹 명세서 파일을 나타낸다. 그룹 명세서 파일(370)은 그룹 이름(372)을 어떠한 수의 멤버(374)에 관련시키는 테이블(371)을 포함한다. 그룹 이름(372)들은 기본적으로 그룹의 멤버(374)를 나타내도록 사용될 수 있는 식별자이다. 일례로서, 가령 그룹 1(372a)의 그룹 이름은 가령 멤버 1(374a) 그리고 멤버 2(374b)와 같은 어떤 수의 멤버와도 관련될 수 있다. 가령 멤버 1(374a)와 같은 한 멤버는 둘이상의 그룹 이름(372)과 관련될 수 있다.

도 4 는 본 발명의 실시예에 따른 자원에 접근하기 위한 요구를 실행하는 방법을 설명하는 처리 흐름도이다. 이같은 처리는 (402)에서 그리고 단계(404)에서 시작되며, 도 1b 의 클라이언트(74)와 같은 요구 클라이언트로부터 도 1b 의 서버(60)와 같은 서버로 호출이 발생하여 도 2a 와 관련하여 상기에서 설명된 바와 같은 적어도 하나의 클래스 파일이다. 도 2b 와 관련하여 상기에서 설명된 바와 같은 보관 파일의 다운로드를 시작하도록 한다. 상기의 요구는 사용자 에이젼트, 즉 가령 앞서 언급한 바의 HotJava

™ 브라우저 또는 Netscape 네비게이터 브라우저와 같은 브라우저를 통해 클라이언트 호출에 응답하여 서버에서 수신된다. 적어도 하나의 클래스 파일 또는 보관 파일 어느 하나의 다운로드 시작이 한 자원에 접근하기 위해 한 요구에 응답하여 일어나며, 따라서 이는 한 애플릿을 실행하기 위한 호출이다. 한 바람직한 실시예에서, 보관 파일은 JAR 파일이다.

단계(406)에서는, 어느 한 보관 파일이 적재되며, 혹은 클래스 파일이 서버로부터 요구 클라이언트와 관련된 메모리내로 적재된다. 일반적으로, 클래스 파일은 클래스들이 보관 파일내에 있지 않으면, 가령 디지탈식으로 표시되지 않으면 적재되며, 클래스들이 디지탈식으로 표시되면 한 보관파일이 적재된다. 마찬가지로, 상기 보관 파일을 적재함은 클래스 파일을 적재함을 포함한다. 클래스 파일이 메모리내로 적재된 뒤에, 확인 처리가 단계(408)에서 적재된 파일상에서 수행된다. 적재된 보관 파일과 관련된 헤더 서명이 유효한가를 검증하는 처리를 포함하는 확인 처리는 보관 파일이 적재된 경우에 유효하며, 도 5 와 관련하여 하기에서 상세히 설명된다.

단계(410)에서 확인처리이후에, 상기 클래스 파일이 한 애플릿으로 변환된다. 즉, 한 애플릿이 JAR 파일의 일부일 수도 있고 그렇지 않을 수도 있는 적재된 클래스 파일 설명하므로써 메모리내에 발생된다. 일단 애플릿이 발생되면, 상기 애플릿 파일이 단계(412)에서 실행된다. 애플릿의 실행과 관련된 단계는 도 6 과 관련하여 하기에서 설명될 것이다.

도 5 는 본 발명의 실시예에 따르고 클래스 파일(class files)들을 검증하는 작업과 관련된 단계 즉 도 4 의 단계(408)를 설명하는 플로우 선도(flow diagram)이다. 프로세스는 단계(502)에서 시작하고 단계(504)에서 아키브 파일(archive file) 또는 클래스 파일이 적재되었는가에 대해 결정이 이루어진다. 만약 클래스파일이 적재되었다면, 표준 클래스(standard class) 검증이 수행되는 단계(506)로 프로세스는 진행된다. 전형적으로 표준클래스 검증은 클래스파일들내의 어떤 것이 안전(security)과 타협가능한가를 확인하기 위해 적재된 모든 클래스파일들, 따라서 클래스들의 검사(check)를 포함한다. 어떤 실시예들에 있어서 실제 장치, 예를 들어 자바(Java

TM), 실제장치의 안전이 타협될 수 있는 가를 결정하기 위해 검사가 이루어진다. 일반적으로 표준 클래스 검증방법(Standard class verification methods)이 당해업자들에게 공지되어 있다. 일단 표준클래스 검증이 수행되면, 클래스파일들의 검증하는 프로세스가 단계(520)에서 완성된다.

단계(504)에서 이루어지는 결정이 아키브 파일이 적재되는 것이라면, 다음에 단계(508)에서, 아키브 파일의 헤더(header)가 검증(validate)되거나 허가된다. 일반적으로 아키브파일은 헤더 시그너쳐(header signature)를 기본으로 하는 아키브파일의 근원을 확인하는 작업을 포함한다. 즉 헤더시그너쳐 따라서 아키브파일의 근원을 설정하도록 검사가 이루어진다. 또한 검증은 아키브파일과 관련된 데이타가 완전한가의 검사를 포함한다. 어떤 실시예들에 있어서 아키브파일은 헤더 시그너쳐를 포함하지 않을 수도 있음을 인식해야 한다. 실시예들에 의해 인트라넷(intranet) 내부의 아키브파일은 시그너쳐가 없을 수도 있다. 단계(510)에서 헤더가 유효(valid)한가 여부에 대한 결정이 이루어진다. 헤더가 유효하지 않다면, 예를 들어 아키브의 내용이 시그너쳐(signature)와 일치하지 않는다면 다음에 단계(514)에서 에러 플래그(error flag) 또는 유사한 것이 발생된다. 한 개의 실시예에 있어서, 에러 플래그는 버려지는 것이 제외될 수 있다. 다른 실시예에 있어서 에러플래그는 요구하는 클라이언트에게 돌려보내지는 메세지를 발생할 수 있다. 에러플래그가 발생한 후에 클래스파일들을 검증하는 프로세스는 단계(520)에서 종료한다.

헤더가 단계(510)에서 유효하다고 판단되면, 프로세스는 단계(510)에서 부터 아키브파일과 관련된 모든 클래스들이 유효화상태를 유지할 것인지를 결정하는 단계(512)를 이동한다. 유효화되는 클래스가 존재한다면, 다음에 단계(516)에서 표준클래스 검증이 수행된다. 단계(506)에서 앞서 설명한 것처럼 표준클래스 검증은 주어진 클래스내의 모든 것이 실제장치의 안전과 타협가능한가를 검증하는 것을 포함한다. 예를 들어, 주어진 클래스내의 어떤 것이 실제장치상의 파일들 또는 메모리(memory)상에 오버라이트(overwrite)될 수 있다면 실제장치의 안전은 타협될 수 있다. 표준클래스검증이 주어진 클래스상에서 완성된 후에 유효하게 되야할 다른 더 많은 클래스들이 있는가를 결정하는 단계(512)로 프로세스제어가 돌아간다. 다른 추가의 클래스들이 유효화되는 상태를 유지하지 않고, 이때 클래스파일들을 유효화하는 과정은 단계(520)에서 완성되는 결정이 단계(512)에서 이루어질때까지 프로세스제어는 단계(512)와 단계(516) 사이의 루프(loop)내에 있게 된다.

도 6 은 본 발명의 실시예에 따르는 애플릿(applet)을 수행하는 작업과 관련된 단계들을 설명하는 프로세스 플로우선도이다. 즉 도 4 의 단계(412)가 설명된다. 프로세스는 단계(602)에서 시작하고, 단계(604)에서 애플릿이 동작을 수행하는 명령을 포함하는가에 대한 결정이 이루어진다. 일반적으로 동작은 시스템레벨리소스(system-level resource)에 접속하는 콜(call)이 될 수 있다. 만약 애플릿이 동작을 수행하는 명령을 포함하지 않는다면 애플릿을 수행하는 프로세스는 단계(616)에서 종료한다. 애플릿이 동작을 수행하는 명령을 포함하지 않는다면, 프로세스플로우는 수행해야 하는 동작이 보호되는, 예를 들어 안전한 동작인가를 결정하는 단계(606)로 진행한다. 즉 동작은 접속이 제어되는 동작인가에 관한 결정이 이루어진다. 만약 동작이 보호되지 않는다면, 동작은 단계(608)에서 수행되고 프로세스플로우는 다른 동작을 수행하는 명령이 존재하는가를 결정하는 단계(604)로 돌아간다.

단계(606)에서 수행명령을 받은 동작이 보호된다고 결정되면, 프로세스플로우는 애플릿 안전관리자(applet security manager)라 불리는 단계(610)로 이동한다. 안전관리자를 부르는 프로세스는 도 7 을 참고로 하기에서 더욱 상세히 설명된다. 전형적으로 애플릿 안전관리자는 주어진 애플릿들로 접속가능한 동작들을 제어한다. 실시예에 있어서 애플릿 안전관리자는 Java

TM 애플릿 안전관리자이다. 단계(612)에서, 동작이 허용되는가가 결정된다. 달리 말하면, 단계(612)는 애플릿이 수행되어질 동작에 접속하는가를 결정한다. 만약 동작이 허용된다면, 동작은 단계(608)에서 수행된다. 단계(608)로 부터 프로세스제어는 다른 동작을 수행하는 명령이 존재하는 가를 결정하는 단계(604)로 돌아간다.

만약 단계(612)내에서 동작이 허용되지 않는 결정이 이루어진다면, 단계(614)에서 방출되는 익셉숀(exception)을 가지는 것에 의해 수행될 수 있는 에러조건(error condition)이 발생하고, 애플릿을 수행하는 프로세스는 단계(616)에서 종료한다. 어떤 실시예들에서, 익셉션을 방출하는 단계는 방출기능(throw function)을 부르는 과정을 포함할 수 있다는 것을 인식해야 한다. 다른 실시예들에 있어서, 익셉션 방출단계는 사용자 에이전트(user agent)에 의해 표시될 수 있는 에러 메세지(error message)를 전송하는 단계를 포함할 수 있다. 또다른 실시예들에서, 에러처리는 사용자와의 상호작용이 사용자가 애플릿에 의한 동작수행을 허용하는 가를 물어보는 식으로 발생하게 만든다. 상기 실시예에서, 접속파일들은 사용자에 의해 제공되는 응답을 영구적으로 기록하도록 업데이트(update)될 수 있다.

도 7 에 있어서, 안전관리자(security manager)를 콜(call)하는 프로세스 즉 도 6 의 단계(610)가 설명될 것이다. 사용자 에이전트(user agent)는 일반적으로 안전관리자와 관련된 것만을 가진다는 것을 인식해야만 한다. 안전관리자를 부르는 프로세스는 단계(702)에서 시작하고, 단계(704)에서 애플릿에 의해 불려지는 동작이 확인된다. 다수의 동작들중에 하나의 동작이 될 수 있을 지라도, 일반적으로 동작은 판독동작(read operation) 또는 쓰기 동작(write operation)이다. 단계(704)에서 프로세스플로우는 동작과 관련된 리소스의 이름이 확인되는 단계(706)로 진행한다. 어떤 실시예들에 있어서, 리소스의 이름은 안전관리자에 대한 콜(call)로 전달되고 따라서 확인된다. 그러나 리소스의 이름이 콜(call)로 전달되지 않을 때, 도 3a 에 대해 앞서 설명된 것처럼 프로퍼티파일이 관련 리소스를 확인하는데 사용될 수 있다.

일단 관련 리소스가 단계(706)내에 확인된다면, 리소스에 해당하는 접속파일의 이름이 형태파일(configuration file)을 이용하여 확인되고, 이것은 애플릿과 관련하여 도 3b 에 대하여 앞서 설명되었다. 애플릿에 해당하는 허가가 다음에 단계(710)내의 접속파일로 부터 얻어진다. 어떤 실시예들에서 적합한 접속파일은 메모리내의 실제 접속파일의 표현이 될 수 있다는 것을 인식해야 한다. 도 3c 에 관련하여 상기에서 설명된 것처럼, 접속파일은 한 세트의 허가(permission)와 개별적인 호스트(host)들 또는 그룹(group)들을 관련시킨다. 허가를 얻은 후에, 안전관리자에 대한 콜이 단계(712)에서 완성된다.

도 8 은 본 발명을 따르는 전형적인 컴퓨터 시스템을 설명한다. 컴퓨터 시스템(830)은 (중앙처리유니트(CPU)로 일컬어지는) 가능한 수의 프로세서(832)를 포함하고, 상기 프로세서는 (전형적으로 리드 온리 메모리(read only memory), 또는 롬(ROM)) 제 1 저장장치(834)와 (전형적으로 랜덤 액세스 메모리(random access memory) 또는 램(RAM)) 제 2 저장장치(836)를 포함하는 기억장치와 연결한다. 종래기술에서 공지된 것처럼 롬(834)은 단일방향으로 CPU에 데이타 및 명령을 전송하고, 램(836)은 전형적으로 양방향으로 데이타 및 명령을 전송하는데 이용된다. 제 1 저장장치(834, 836) 모두는 상기 설명과 같이 적합한 모든 연산가능한 매체를 포함할 수 있다. 매쓰 기억장치(mass memory device)(838)는 또한 양방향으로 CPU(832)에 연결되고 추가의 데이타 저장용량(data storage capacity)을 제공한다. 대량 기억장치(838)는 프로그램, 데이타 및 유사물을 저장하는데 이용가능하고 전형적으로 제 1 저장장치(834, 836)보다 느린 하드 디스크(hard disk)와 같은 제 2 저장 매체(secondary storage medium)가 된다. 대량 메모리 저장장치(838)는 자기 또는 종이 테입리더(magnetic or paper tape reader) 또는 다른 공지 장치의 형태일 수 있다. 적절한 경우에 있어서, 매쓰 메모리 장치(838)내에 유지되는 정보는 실제 메모리로서 램(836)의 일부로서 표준형태로 구성된다. CD-ROM(834)와 같은 특정의 매쓰저장장치는 또한 일방향으로 CPU에 데이타를 보낸다.

CPU(832)는 또한, 한정되지는 않더라도, 비디오 모니터(video monitor), 트랙볼(track ball), 마우스, 키보드(keyboard), 마이크로 폰, 터치 감지 디스플레이(touch-sensitive display), 트랜스듀서 카드리더(transducer card reader), 자기 또는 종이 테입 리더(magnetic or paper tape reader), 태블렛(tablet), 스타일러스(styluses), 음성 또는 수기 인식장치(voice or handwriting recognizer) 또는 다른 공지 입력장치 물론 다른 컴퓨터와 같은 장치를 포함하는 한 개 또는 한 개 이상의 입력/출력장치(840)에 연결된다. 마지막으로 선택적으로 CPU(832)는 네트워크 연결장치(812)를 이용하여 인터넷 네트워크 또는 인트라넷과 같은 컴퓨터 또는 텔레커뮤니케이션 네트워크에 연결될 수 있다. 상기 네트워크 연결장치를 가지고, CPU는 네트워크로 부터 정보를 받고 상기 방법단계를 수행하는 과정에서 네트워크에 정보를 보낼 수 있다는 것을 알게 된다. 상기 장치 및 재료는 컴퓨터 하드웨어 및 소프트웨어 기술분야의 당해업자들에게 친숙할 것이다. 추가로 네트워킹 장치 및 상기 하드웨어 및 소프트웨어 요소가 표준설계 및 구성으로 이루어짐을 당해업자들은 인식해야 한다.

여기서 설명되는 컴퓨터 수행방법은 컴퓨터 시스템상에서 컴퓨터프로그램을 수행하기 위해 컴퓨터학 분야에서 공지된 기술 및 장치를 이용하여 수행될 수 있다. 여기서 이용되는 것처럼 컴퓨터 시스템(computer system)이라는 용어는 제한적이지는 않지만 램(RAM), 롬(ROM), CD롬(CD-ROM) 및 유사장치를 포함하고 프로세싱 유니트(processing unit)와 데이타 및 명령을 교환하기 위한 한 개 또는 한 개 이상의 데이타 저장장치와 연결되는 데이타 및 명령을 처리하기 위한 (중앙처리유니트, CPU) 처리장치를 포함함을 나타낸다. 데이타 저장장치는 컴퓨터네트워크에 대해 처리유니트와 연결되는 처리유니트와 직접 연결될 수 있다. 컴퓨터 네트워크에 대해 처리유니트에 연결되는 원격 데이타 저장장치들은 특별한 워크스테이션(workstation)에서 처리하기 위한 처리유니트에 대한 프로그램 명령을 감지할 수 있음을 인식해야 한다. 추가로, 처리유니트는 (예를 들어 분산된 프로세서) 컴퓨터네트워크에 대해 또는 (예를 들어 병렬식 프로세서) 동일한 물리적구조를 통해 한 개 또는 한 개이상의 추가 처리장치들과 연결될 수 있다. 원격으로 연결된 상기 데이타 저장장치 및 프로세서의 이용은 컴퓨터학 분야의 당해업자들에게 익숙할 것이다(Ralston 1993을 참고). 여기서 이용되는 것처럼 컴퓨터 네트워크(computer network)라는 용어는 서로 송수신이 이루어질 수 있는 한 세트의 컴퓨터 시스템들을 상호결합시키는 한 세트의 커뮤니케이션 채널(communication channel)을 포함하는 것을 나타낸다. 커뮤니케이션 채널은 제한적이지는 않지만 비틀린 쌍의 와이어(twisted pair wire), 동심케이블(coaxial cable), 광섬유(optical fiber), 위성연결(satellite links) 또는 디지탈 마이크로 라디오(digital microwave radio)와 같은 전송매체를 포함한다. 컴퓨터 시스템은 넓은 또는 와이드(wide) 영역(예를 들어 수십, 수백 또는 수천 마일, WAN), 또는 국소영역네트워크(예를 들어 수피트에서 수백피트에 걸친 LAN)에 걸쳐 분포될 수 있다. 추가로 다양한 국소영역(local-area) 및 광역(wide-area) 네트워크가 컴퓨터시스템의 집합 네트워크를 형성하도록 결합될 수 있다. 컴퓨터 네트워크의 상기 집합의 한 예가 인터넷(internet)이다.

발명의 효과

단지 본 발명의 몇가지 실시예들이 설명되었을지라도, 본 발명은 본 발명의 사상 또는 범위로부터 벗어나지 않고 다수의 다른 특정예들에서 실시될 수 있다는 것을 이해해야만 한다. 예를 들어, 시그너쳐를 받는 아키브파일데이타의 오직 한 개의 형태가 설명되었을지라도 상기 아키브 파일 데이타구조는 본 발명의 범위내에서 다양하게 변화될 수 있음을 인식해야 한다. 추가로 접속시스템 리소스들에 대해 리퀘스트(request)를 수행하는 방법과 관련된 단계들이 기록될 수 있다. 또한 본 발명의 사상 또는 범위로부터 벗어나지 않고 단계들이 제거 및 추가될 수 있다. 그러므로 설명된 실시예들은 설명적이지 제한적인 성격을 갖지 않는 것으로 받아들여져야 하고 본 발명은 등가한 것들의 다음 청구항들 및 모든 범위에 의해 정의되어야 한다.

(57)청구의 범위

청구항1

(a) 소프트웨어 프로그램을 위한 운영체제자원으로의 접근 정도를 규정하고,

(b) 상기 소프트웨어 프로그램이 상기 컴퓨터에 의해 실행되는때 상기 소프트웨어 프로그램내 이용될 수 있는 시스템-수준 접근 정도를 결정하기 위해 상기 소프트웨어 프로그램과 관련된 적어도 하나의 파일을 시험하며,

(c) 상기 컴퓨터에서 상기 소프트웨어 프로그램을 실행하고,

(d) 상기 소프트웨어 프로그램이 상기 컴퓨터에서 실행되는 때 상기 소프트웨어 프로그램과 관련된 프로그램 지시를 인터셉트하며,

(e) 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 바깥쪽인 동작을 포함하는가를 결정하고, 그리고

(f) 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도내에 있는 상기 컴퓨터와 관련된 시스템-수준 자원에 접근하기 위한 허가를 얻게 되는때 상기 프로그램 지시를 수행하는 단계를 포함하는,

상기 운영체제를 실행하는 컴퓨터에서 사용되는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항2

제 1 항에 있어서, 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도를 벗어나는 동작을 포함하는가를 결정하는 단계가 상기 소프트웨어 프로그램과 관련된 식별자를 유효하게 함을 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항3

제 1 항 또는 2 항에 있어서, 상기 프로그램 지시를 실행하는 단계가 상기 프로그램 지시에 의해 접근되는 상기 시스템-수준자원이 시스템-수준 자원에 의해 보호되는가를 결정함을 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항4

전술한 항중 어느 한 항에 있어서, 상기 소프트웨어 프로그램이 한 애플릿을 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항5

제 4 항에 있어서, 상기 애플릿이 자바 애플릿임을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항6

제 4 항 또는 5 항에 있어서, 상기 애플릿이 헤더와 관련되며, 상기 헤더가 식별자를 포함하도록 되고, 상기 식별자는 상기 파일의 근원을 나타내도록 배열됨을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항7

제 6 항에 있어서, 상기 컴퓨터가 상기 시스템-수준 자원에 접근하기 위한 허가를 얻었는가를 결정하기 위해 상기 식별자를 확인하는 단계를 더욱더 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항8

제 4-7 항중 어느 한 항에 있어서, 상기 컴퓨터가 클라이언트 컴퓨터이고, 상기 애플릿이 서버 컴퓨터로부터 상기 클라이언트 컴퓨터로 다운로드됨을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항9

제 8 항에 있어서,

(a) 상기 시험하는 단계가 상기 애플릿에 대한 서버 컴퓨터와 관련된 시스템-수준 자원으로의 접근 정도를 규정하므로써 규정되는 상기 클라이언트에 의해 실행되는때 상기 애플릿에 이용될 수 있는 서버로의 시스템-수준 접근정도를 결정함을 포함하며,

(b) 상기 결정단계가 상기 서버와 관련된 시스템-수준 자원에 접근하기 위한 상기 프로그램 지시가 상기 애플릿에 이용될 수 있는 시스템-수준 접근 정도를 벗어나는 동작을 포함하고,

(c) 상기 실행하는 단계가 상기 프로그램 지시를 실행함을 포함하여 상기 애플릿에 이용될 수 있는 시스템-수준 접근 정도내에 있는 상기 서버 컴퓨터와 관련된 시스템-수준 자원에 접근하는 것이 상기 애플릿에 허용되는때 상기 서버 컴퓨터와 관련된 시스템-수준 자원에 접근하도록 함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항10

(a) 소프트웨어 프로그램을 위한 운영체제자원으로의 접근 정도를 규정하고,

(b) 상기 클라이언트 컴퓨터에서 상기 소프트웨어 프로그램을 실행하기 위한 지시를 포함하는 적어도 하나의 파일을 적재하며,

(c) 상기 소프트웨어 프로그램이 상기 컴퓨터에 의해 실행되는때 상기 소프트웨어 프로그램내 이용될 수 있는 시스템-수준 접근 정도를 결정하기 위해 상기 접근 정도를 규정하는 단계에 의해 규정된 바와 같이 적어도 하나의 파일을 시험하며,

(d) 상기 클라이언트 컴퓨터에서 상기 소프트웨어 프로그램을 실행하고,

(e) 상기 소프트웨어 프로그램이 상기 클라이언트 컴퓨터에서 실행되는 때 상기 소프트웨어 프로그램과 관련된 프로그램 지시를 인터셉트하며,

(f) 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 바깥쪽인 동작을 포함하는가를 결정하고, 그리고

(g) 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도내에 있는 시스템-수준 자원에 접근하기 위한 허가를 얻게 되는때 상기 프로그램 지시를 수행하는 단계를 포함하는,

적어도 몇 개의 운영체제 자원이 컴퓨터 네트워크를 통해 상기 클라이언트 컴퓨터에 결합되는 서버 컴퓨터에 존재하며,

상기 운영체제를 실행하는 클라이언트 컴퓨터에서 사용되는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항11

제 10 항에 있어서, 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도를 벗어나는 동작을 포함하는가를 결정하는 단계가 상기 소프트웨어 프로그램과 관련된 식별자를 유효하게 함을 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항12

제 10 항 또는 11 항에 있어서, 상기 프로그램 지시를 실행하는 단계가 상기 프로그램 지시에 의해 접근되는 상기 시스템-수준자원이 시스템-수준 자원에 의해 보호되는가를 결정함을 포함함을 특징으로 하는 소프트웨어 프로그램을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항13

제 10-12 항중 한 항에 있어서, 상기 컴퓨터 네트워크에서 상기 클라이언트 컴퓨터와 상기 서버 컴퓨터사이의 자료 전달 통신링크를 설정하며, 그리고

상기 컴퓨터 네트워크에서 상기 서버 컴퓨터로부터 상기 클라이언트 컴퓨터로 적어도 하나의 파일을 전송하는 단계를 더욱더 포함함을 특징으로 하는 소프트웨어 프로그램을 위한 운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항14

(a) 상기 요구와 관련된 파일의 다운로드를 시작하기 위해 두 번째 서버를 호출하고,

(b) 상기 관련 파일들을 상기 두 번째 서버로부터 적재하며, 상기 관련 파일들이 한 보관 파일을 포함하고, 상기 보관 파일이 적어도 하나의 클래스 파일과 한 헤더를 포함하며, 상기 헤더가 상기 보관 파일의 기원을 나타내도록 된 식별자를 포함하고,

(c) 상기 보관 파일을 확인하며,

(d) 상기 클래스 파일을 한 애플릿으로 변환시키고, 그리고

(e) 상기 애플릿을 실행하며, 상기 애플릿은 적어도 하나의 지시를 포함하는 단계를 포함하며,

상기 애플릿을 실행하는 것이 상기 클라이언트가 상기 첫 번째 서버와 관련된 상기 시스템 자원에 접근할 수 있도록 하는 첫 번째 서버와 관련된 시스템 자원에 접근하기 위해 한 클라이언트로부터의 요구를 처리하기 위한 방법.

청구항15

제 14 항에 있어서, 상기 보관 파일을 확인하기 위한 단계가

(a) 상기 헤더에 권한을 주고,

(b) 상기 헤더가 유효한가를 결정하며,

(c) 상기 헤더가 유효함이 결정되는때 상기 클래스에서의 클래스 검증을 수행하는 서브-스텝을 포함함을 특징으로 하는 방법.

청구항16

제 14 항 또는 15 항에 있어서, 상기 애플릿을 실행하는 단계가

(a) 상기 지시가 보호된 동작을 실행하기 위한 지시임을 결정하고,

(b) 상기 지시가 보호된 동작을 실행하기 위한 지시가 아님이 결정된때 상기 동작을 실행시키며, 그리고

(c) 상기 지시가 보호된 동작을 실행하기 위한 지시임이 결정되는때 상기 동작이 허용되는가를 결정하는 서브-스텝을 포함함을 특징으로 하는 방법.

청구항17

소프트웨어 프로그램을 실행하기 위한 지시를 포함하는 적어도 하나의 파일을 보유하는 적어도 하나의 메모리에 결합된 첫 번째 컴퓨터를 포함하며,

상기 소프트웨어 프로그램은 상기 첫 번째 컴퓨터에서 실행되고, 상기 첫 번째 컴퓨터가 상기 운영체제를 실행시키고,

상기 첫 번째 컴퓨터가

(a) 소프트웨어 프로그램을 위한 운영체제자원으로의 접근 정도를 규정하고, 상기 첫 번째 컴퓨터가 한 프로그램 지시가 상기 소프트웨어와 관련이 있는가를 결정하도록 하며,

(b) 상기 첫 번째 컴퓨터에 상기 소프트웨어 프로그램을 실행하기 위한 지시를 포함하는 적어도 하나의 파일을 적재하며,

(c) 상기 소프트웨어 프로그램이 상기 첫 번째 컴퓨터에 의해 실행되는때 상기 소프트웨어 프로그램내 이용될 수 있는 시스템-수준 접근 정도를 결정하기 위해 상기 적어도 하나의 파일을 시험하며,

(d) 상기 첫 번째 컴퓨터에서 상기 소프트웨어 프로그램을 실행하고,

(e) 상기 소프트웨어 프로그램이 상기 첫 번째 컴퓨터에서 실행되는 때 상기 소프트웨어 프로그램과 관련된 프로그램 지시를 인터셉트하며,

(f) 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 바깥쪽인 동작을 포함하는가를 결정하고, 그리고

(g) 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도내에 있는 상기 첫 번째 컴퓨터와 관련된 시스템-수준 자원에 접근하기 위한 허가를 얻게 되는때 상기 프로그램 지시를 수행하도록 구성되는,

운영체제 자원으로의 접근 정도를 제어하는 방법.

청구항18

제 17 항에 있어서, 상기 첫 번째 컴퓨터가 상기 프로그램 지시에 의해 접근되는 시스템-수준 자원이 시스템-수준 자원으로 보호되는 가를 결정하도록 됨을 특징으로 하는 방법.

청구항19

(a) 소프트웨어 프로그램을 위한 운영체제자원으로의 접근 정도를 규정하고,

(b) 상기 소프트웨어 프로그램이 상기 컴퓨터에 의해 실행되는때 상기 소프트웨어 프로그램내 이용될 수 있는 시스템-수준 접근 정도를 결정하기 위해 상기 소프트웨어 프로그램과 관련된 적어도 하나의 파일을 시험하며,

(c) 상기 컴퓨터에서 상기 소프트웨어 프로그램을 실행하고,

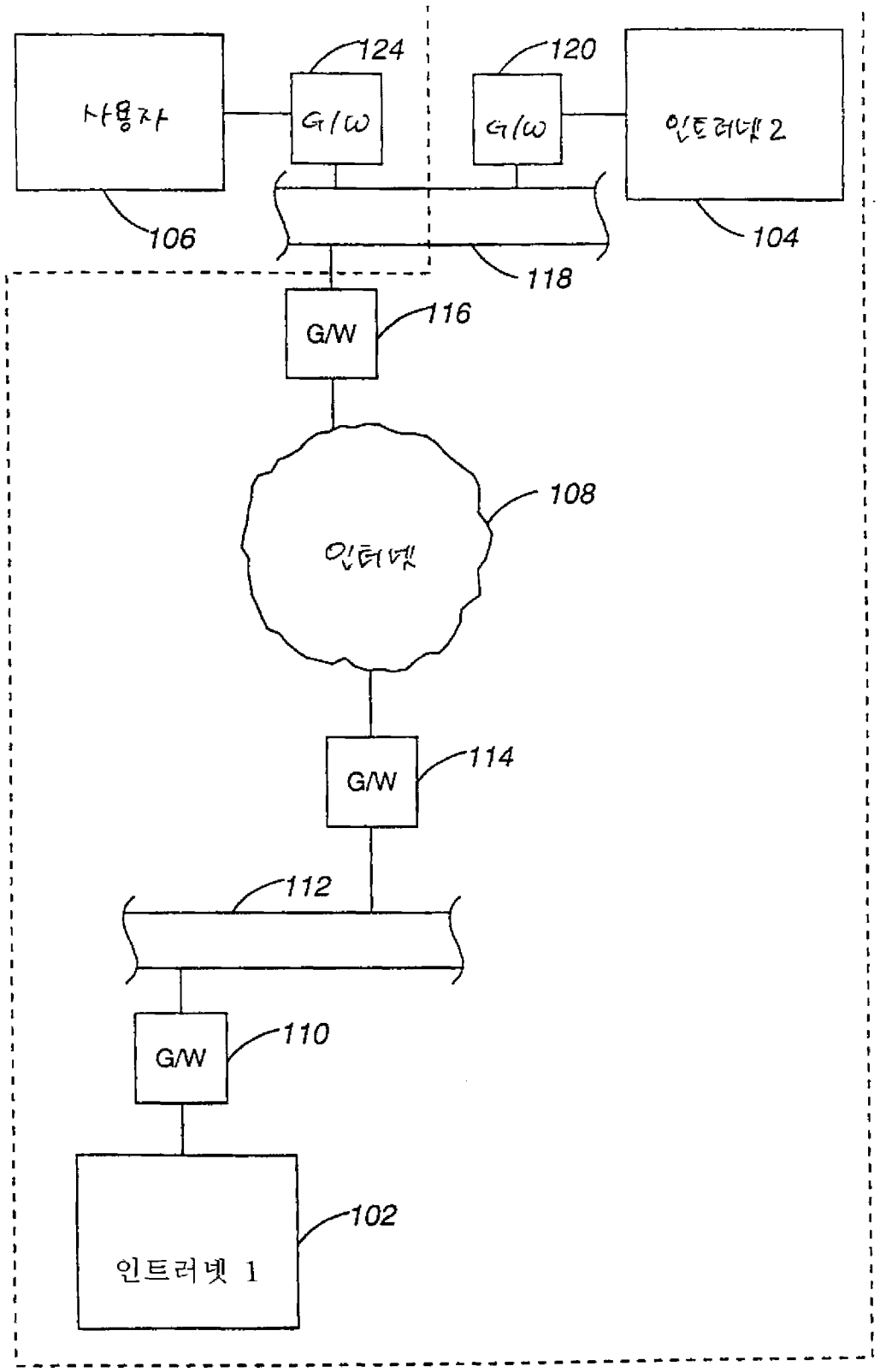
(d) 상기 소프트웨어 프로그램이 상기 컴퓨터에서 실행되는 때 상기 소프트웨어 프로그램과 관련된 프로그램 지시를 인터셉트하며,

(e) 상기 프로그램 지시가 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 바깥쪽인 동작을 포함하는가를 결정하고, 그리고

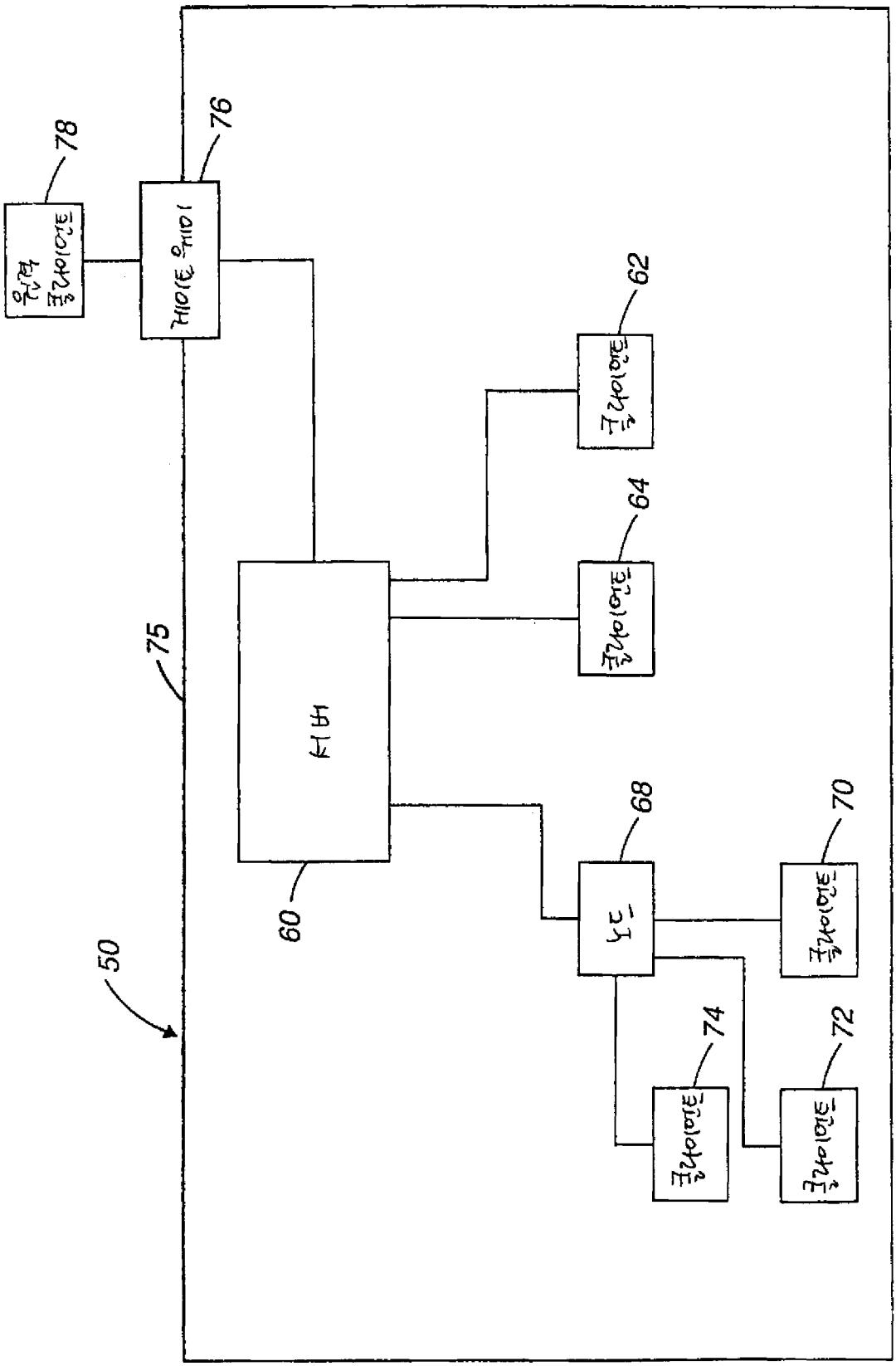
(f) 상기 소프트웨어 프로그램에 이용될 수 있는 시스템-수준 접근 정도내에 있는 상기 컴퓨터와 관련된 시스템-수준 자원에 접근하기 위한 허가를 얻게 되는때 상기 프로그램 지시를 수행하는 작용을 컴퓨터가 수행하도록 구성된 컴퓨터-판독가능 프로그램 코드를 포함하는 컴퓨터 판독가능 매체.

도면

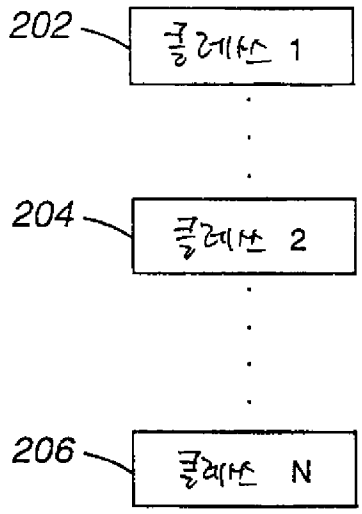
도면1


사용자
124
G/W
120
G/W
인트라넷 2
106
104
118
G/W
116
인터넷
108
G/W
114
112
G/W
110
인트러넷 1
102

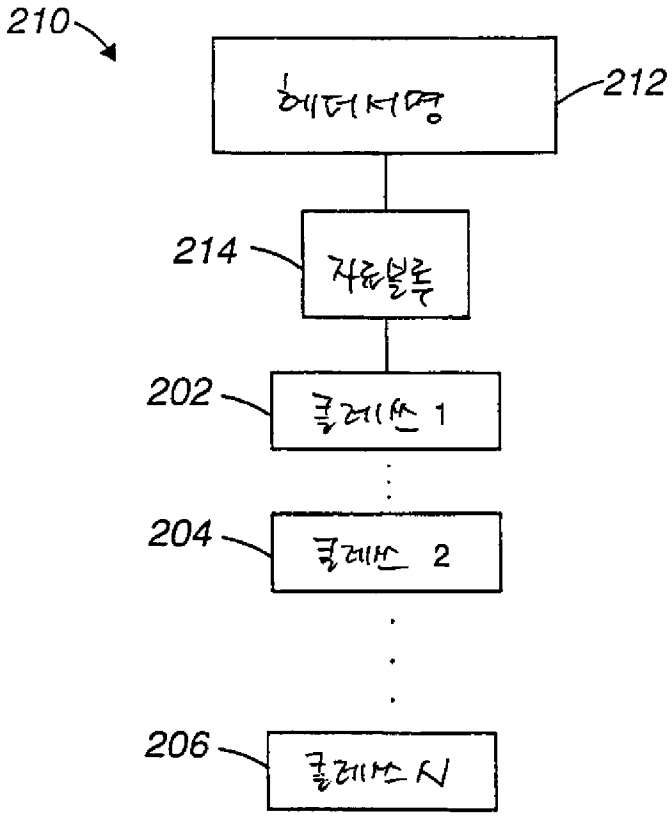
도면1b



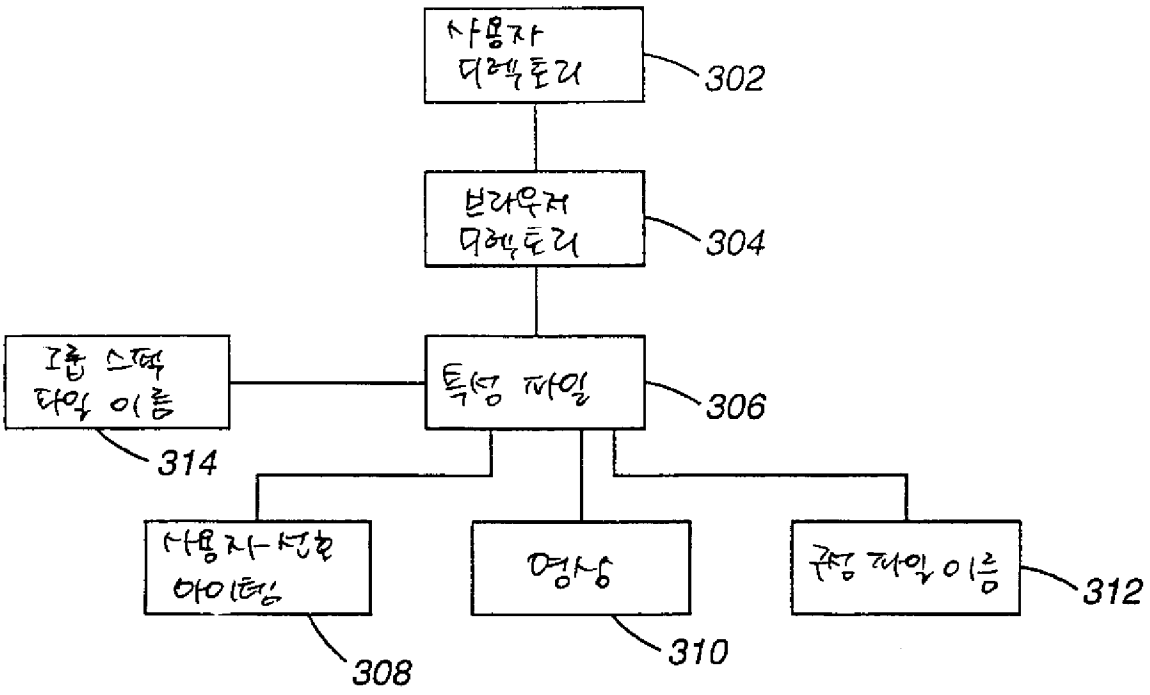
도면2a


202
클래스 1
204
클래스 2
206
클래스 N

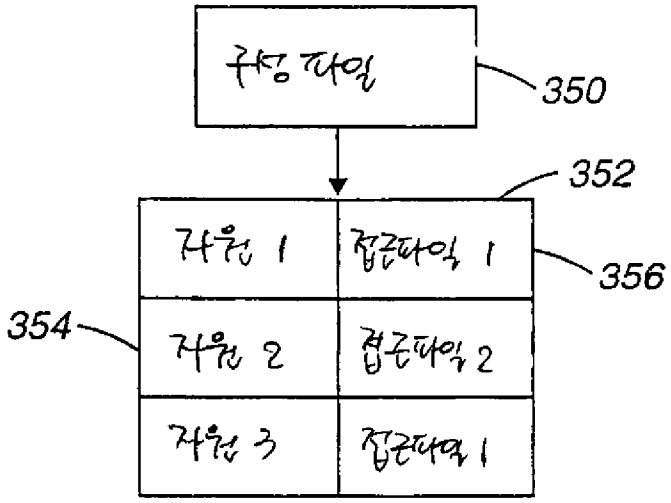
도면2b


210
헤더서명
212
214
자원할당
202
클래스 1
204
클래스 2
206
클래스 N

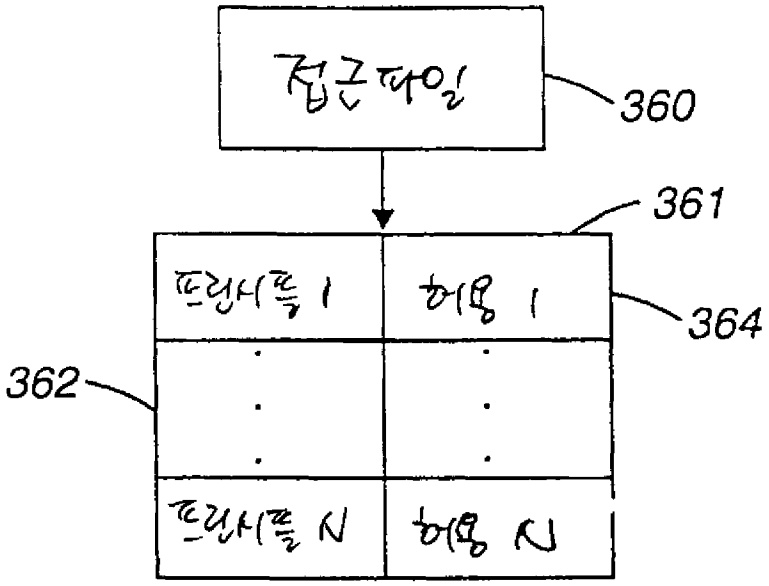
도면3a


사용자 디렉토리
302
브라우저 디렉토리
304
특정 파일
306
구성 스택 파일 이름
314
사용자-정의 아이템
308
영상
310
구성 파일 이름
312

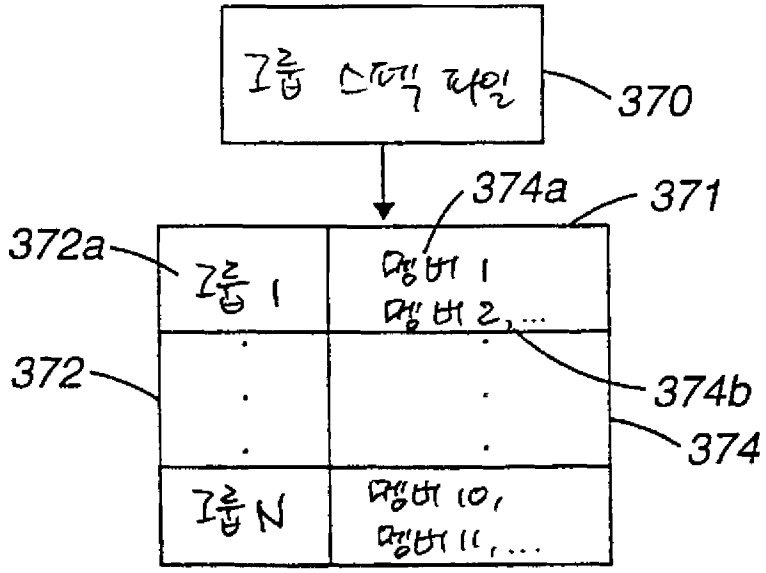
도면3b


구성 파일
350
352
자원 1
접근파일 1
356
354
자원 2
접근파일 2
자원 3
접근파일 1

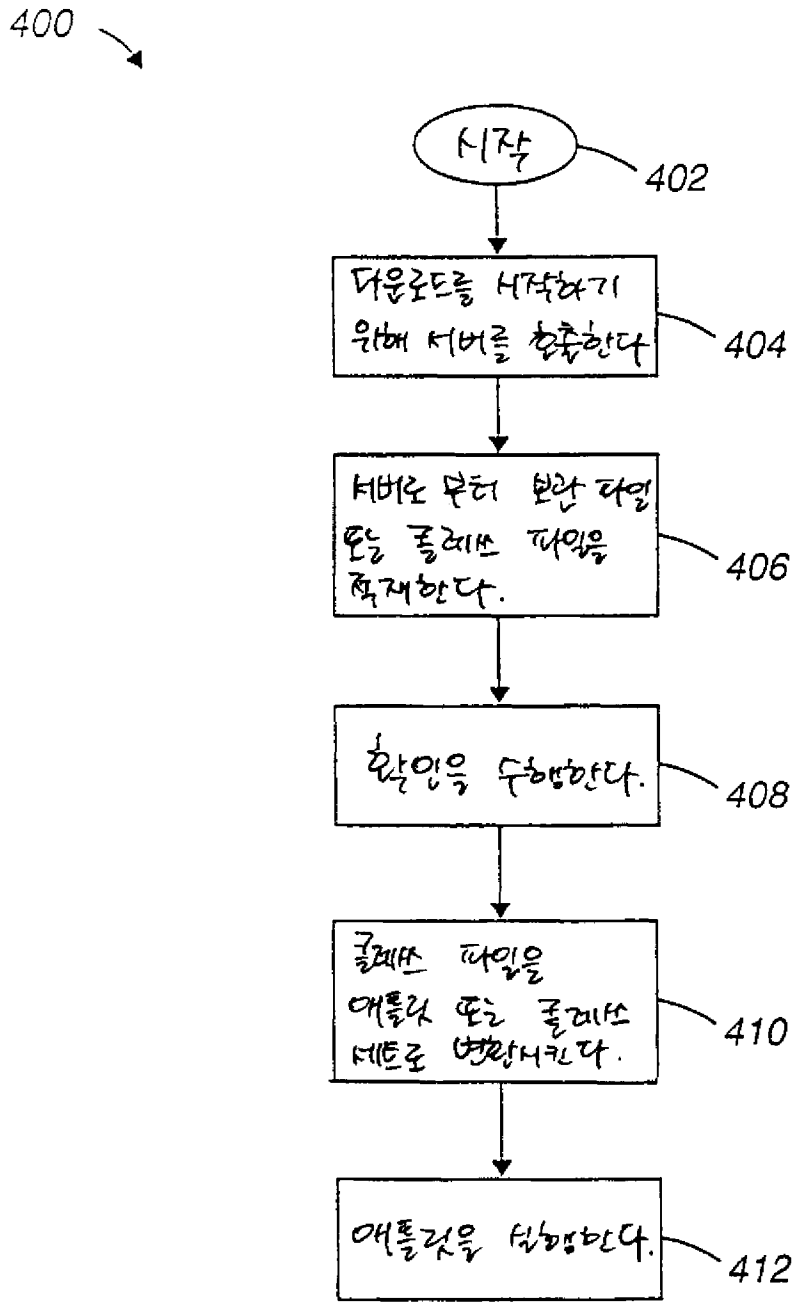
도면3c


접근 제어 파일
360
361
프로세스 1
허용 1
364
362
프로세스 N
허용 N

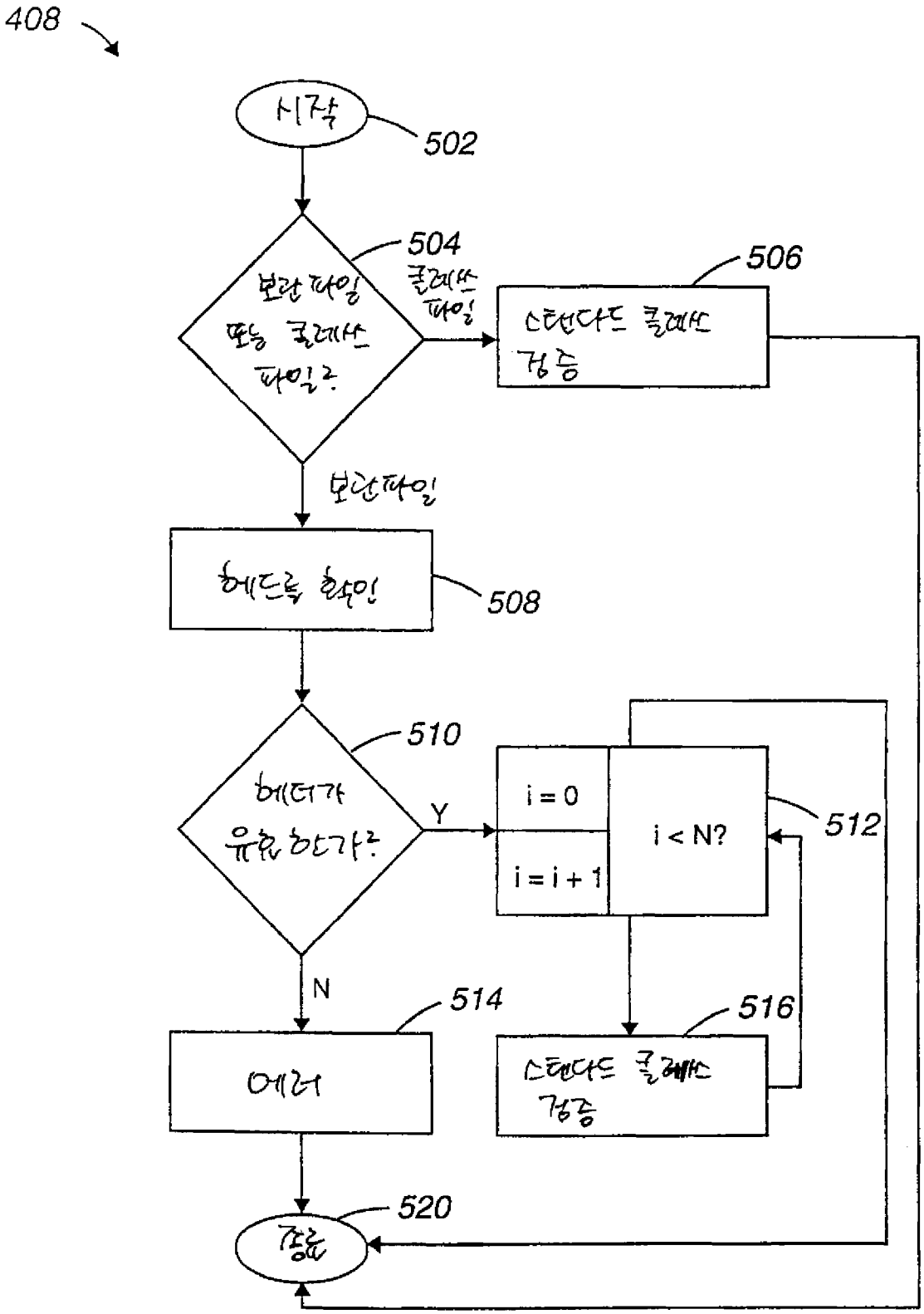
도면3d


그룹 스위칭 파일
370
374a
371
372a
그룹 1
멤버 1
멤버 2, ...
372
374b
374
그룹 N
멤버 10,
멤버 11, ...

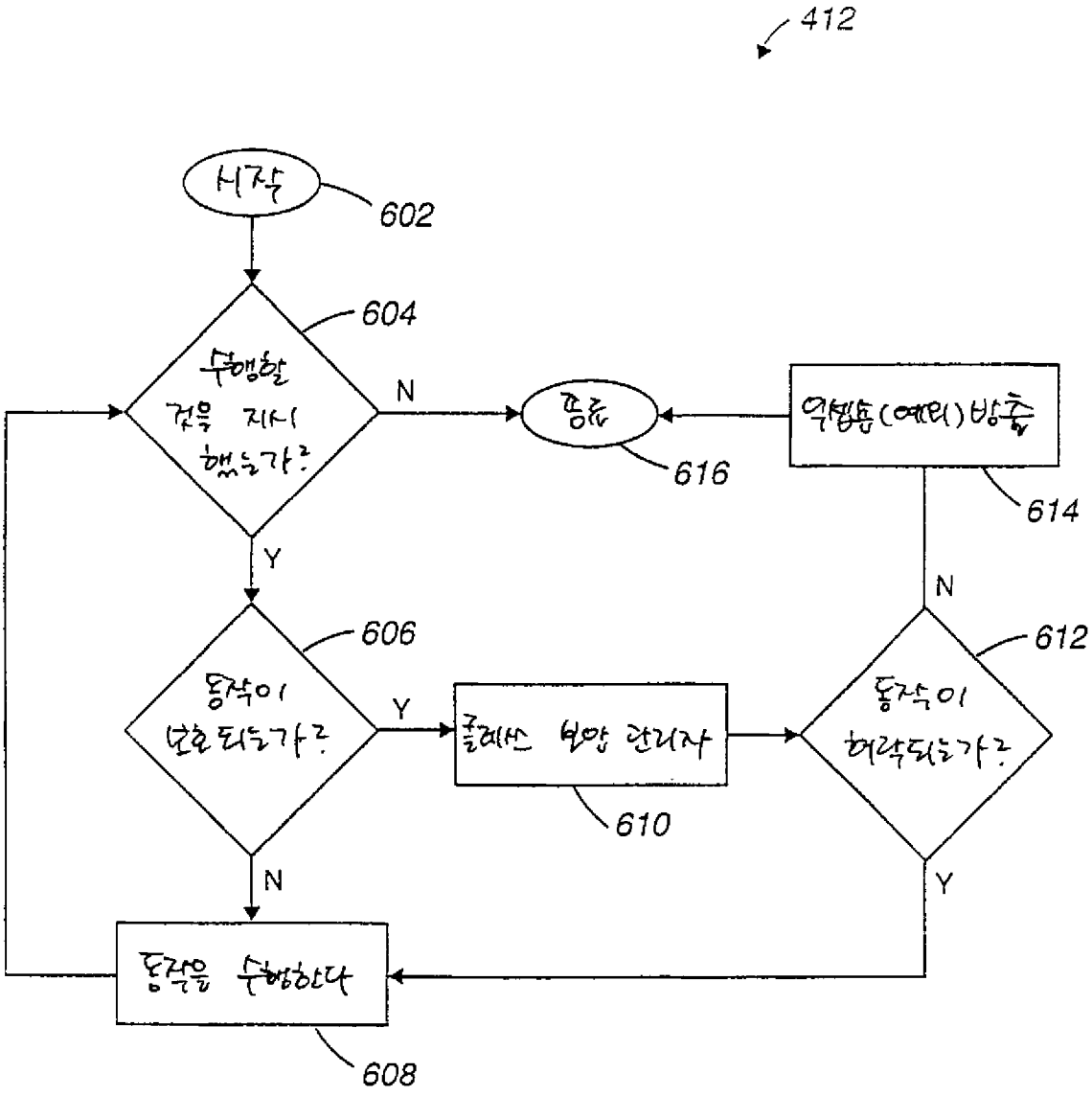
도면4


400
시작
402
다운로드를 시작하기 위해 서버를 호출한다
404
서버로 부터 보안 타일 또는 클래스 파일을 적재한다.
406
확인을 수행한다.
408
클래스 파일을 애플릿 또는 클래스 세트로 변환시킨다.
410
애플릿을 실행한다.
412

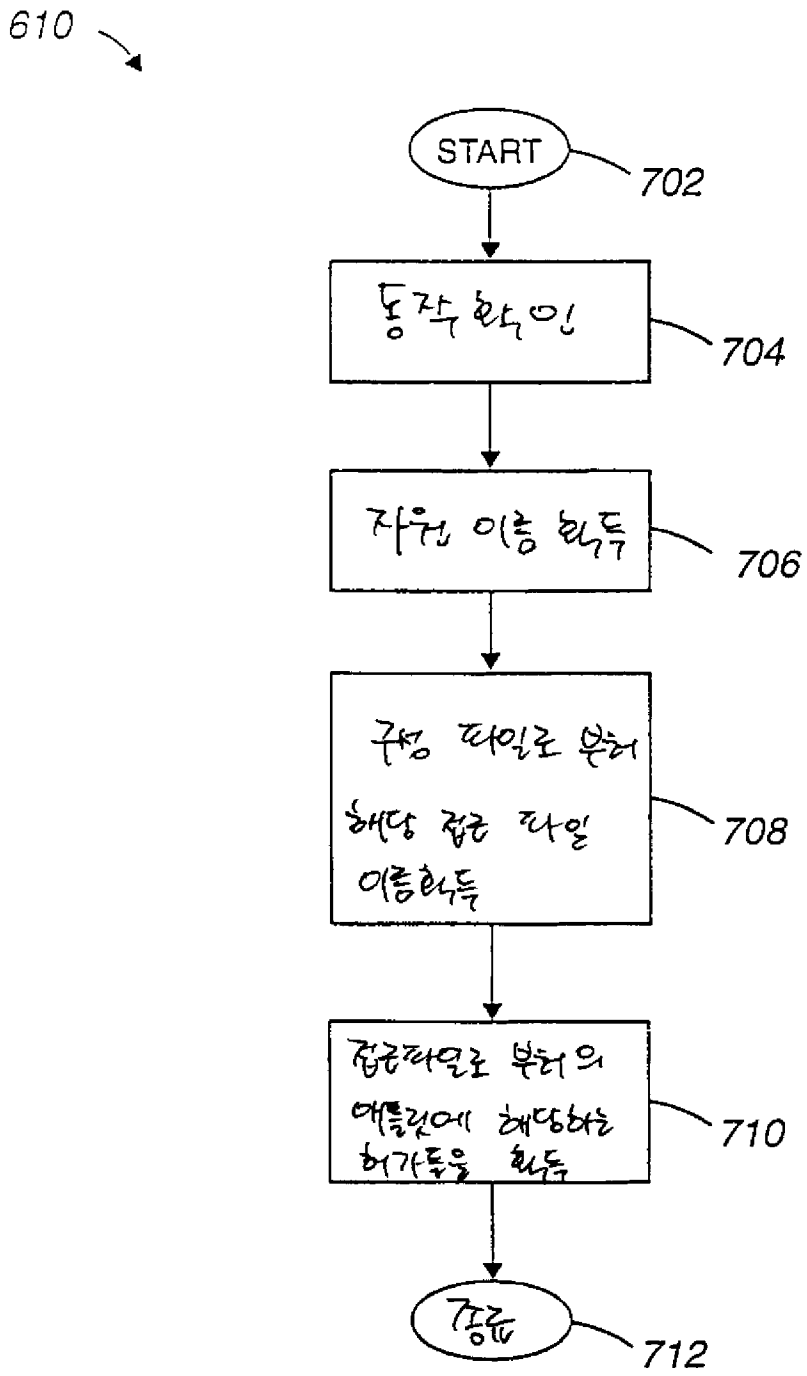
도면5


408
시작
502
보안 파일 또는 클래스 파일?
504
클래스 파일
스탠다드 클래스 검증
506
보안파일
헤더를 확인
508
헤더가 유효하다가?
510
Y
i = 0
i = i + 1
i < N?
512
N
에러
514
스탠다드 클래스 검증
516
종료
520

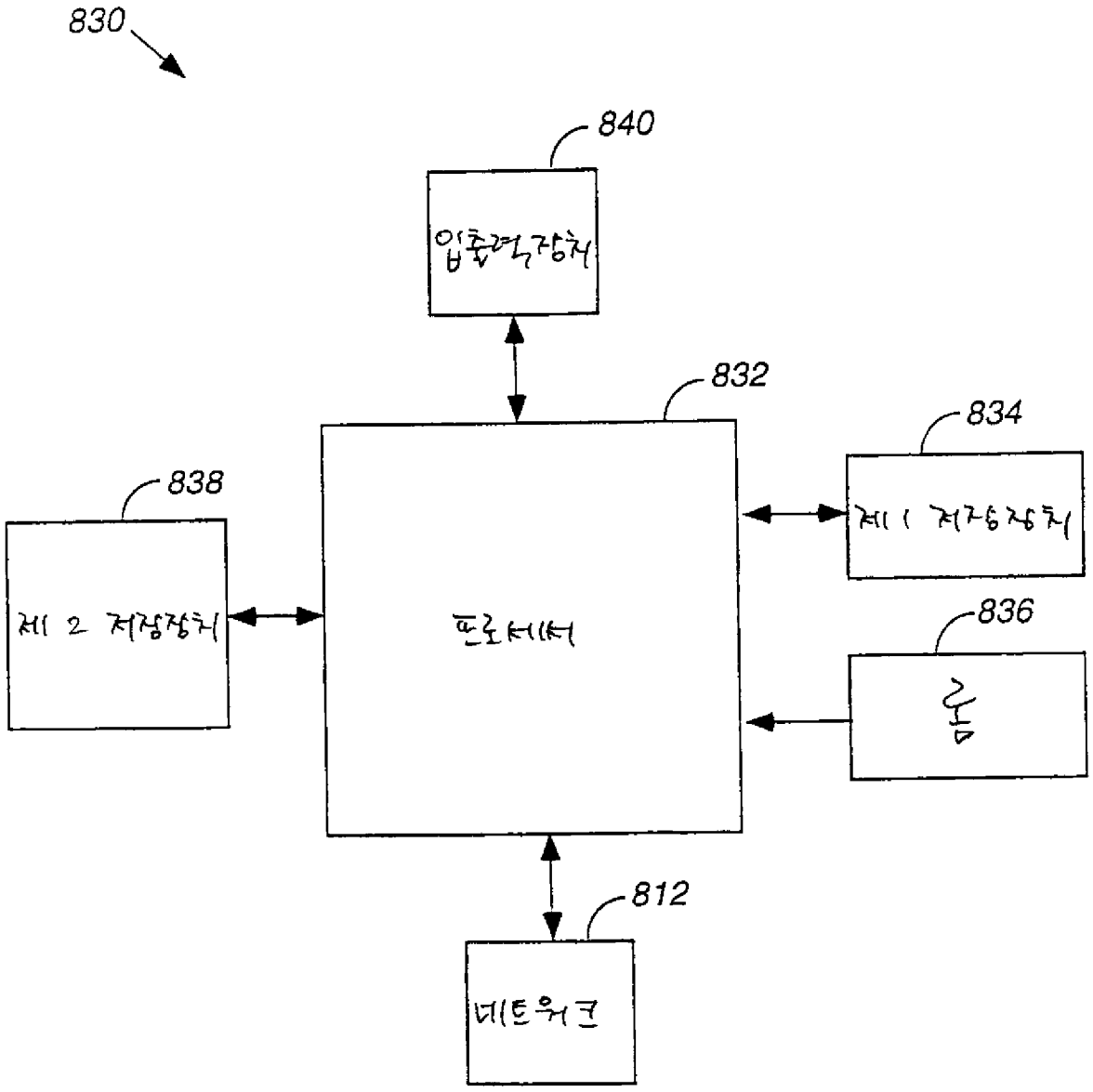
도면6


412
시작
602
604
수행할 것을 지시했는가?
N
종료
616
악성동작(예외) 방출
614
Y
606
동작이 보호되는가?
Y
클래스 보안 관리자
610
612
동작이 허락되는가?
N
Y
동작을 수행한다
608

도면7


610
START
702
동작 확인
704
자원 이름 획득
706
구성 파일로 부터
해당 접근 파일
이름획득
708
접근파일로 부터의
애플릿에 해당하는
허가특을 획득
710
종료
712

도면8


830
840
입출력장치
832
834
838
제 1 저장장치
제 2 저장장치
프로세서
836
812
네트워크